Notes for Meeting 13
State-Driven Problem Solving

A Definition of Problem Solving

We have seen how a cognitive system can use procedural knowledge to handle familiar tasks, but not how it handles NOVEL tasks.

We can formulate the abstract task of PROBLEM SOLVING as:

- Given: A current situation for a real or imagined environment;

- Given: A specification for some desired situation; and

- Given: A set of operations for changing situations and constraints on their use;

- Generate: A situation that satisfies the specification and (optionally) a sequence of operations that produces it.

The ability to solve UNFAMILIAR problems of this sort is one of the hallmarks of human intelligence.

Studies of problem solving have played a central role in the development of AI and cognitive science.

Problem Solving and Planning

The tasks of problem solving and plan generation are similar but not identical in that the former:

- does not always involve physical actions over time

- does not require creation of complete plans

- does not need to occur solely in the agent’s mind

Thus, planning is an important type of problem solving but not the only form.

Another important class of problem-solving tasks involve DESIGN.

Three Hypotheses about Problem Solving

As we have seen, Newell and Simon (1976) make important claims about the nature of intelligent systems:

1. The physical symbol system hypothesis. Physical symbol systems are necessary and sufficient for intelligent behavior.

2. The heuristic search hypothesis. Problem solving involves:

   a. Search through a space of symbolic structures that represent candidate solutions;

   b. Search is modulated by heuristics that guide the process in promising directions.

These assumptions form the basis for many AI systems and computational models of human cognition.

The Notion of a Problem Space

Newell and Simon’s second claim revolves around the idea of a PROBLEM SPACE, which one can specify in terms of:

(a) a representation for candidate states;

(b) an initial state from which to start the search;

(c) operators for producing new states from existing ones;

(d) a test for whether a given state solves the problem.

Together, these elements implicitly define a set of possible states, one of which may satisfy the test and solve the problem.

This basic scheme can be used in many different ways that depend on choices for these four dimensions.

Heuristics for Problem Space Search

Most problem spaces are combinatorial in nature, with the number of candidate states growing exponentially with search tree depth.

However, a problem solver can use HEURISTICS to guide this search; these can take one of two forms:

- Symbolic rules that specify when to prefer, select, or reject an an operator, goal, or state;

- Numeric evaluation functions that the relative desirability of states, goals, or operators.

Both types of heuristics are typically specific to application domain.

E.g., chess masters are good at chess because of domain knowledge that lets them focus heuristically on a small set of moves.

Dimensions of Problem Space Search

Approaches to problem solving differ along three important dimensions:

- Direction of search (forward from states, backward from goals)
- Search regimen (e.g., depth first, breadth first, greedy, sampling)
- Techniques used to select states, goals, and operators

One must also decide whether the problem space includes:

- partial state that are not candidate solutions
- only complete states that are candidate solutions

The latter is referred to as repair-space search or "local" search.

Alternative Search Regimens

One can organize problem-solving behavior using different schemes:

- heuristic depth-first search
- best-first search (including A*)
- heuristic beam search
- greedy search
- iterative deepening
- iterative sampling

These techniques make different demands on memory and explore the search space in distinct ways.

Most bear little relation to problem solving as observed in humans. E.g., chess players use a method known as PROGRESSIVE DEEPENING.

State-Driven Heuristic Search

Much of the work on heuristic problem solving adopts a state-driven approach that carries our forward chaining search. This involves:

- Selecting a state to chain off

- Finding all operator instances applicable in that state

- EITHER

  - Applying these operators to generate successor states OR

  - Selecting an operator and applying it to generate a new state

The first scheme is standard in most game-playing systems, while the second is common in models of human problem solving.

Representation and Inference in Problem Solving

Most AI courses introduce heuristic search early on and do not relate it to other key ideas, but human problem solving relies on:

- Representing sates as relations among entities;

- Making inferences about higher-order relations; and

- Matching relational operator conditions against states.

Problem solving in cognitive systems builds on more basic abilities like conceptual inference and execution.

Production Systems and Heuristic Search

Production systems can be used for routine tasks, but they also offer natural support for state-driven problem solving.

- Rules can specify legal conditions on operators and their effects.

- Relational pattern matching determines how to instantiate rules against the current state.

- Conflict resolution strategies like recency can produce depth-first search control.

Adding appropriate conditions to rules can reduce or eliminate search, suggesting an approach to learning heuristic knowledge.

One can also adapt production systems to support goal-driven problem solving, but this involve writing rules "backwards".

Game Playing as Problem Solving

Research on playing board games has been central to AI and cognitive science since their inception for a number of reasons:

a. Game playing concerns agency, since it involves goal-directed behavior in a dynamic domain that changes in unpredictable ways.

b. Games rules are simple and abstract, yet they generate a large variety of situations that challenge even accomplished players.

c. Thus, they require an interesting combination of knowledge and search, two factors that play key roles in cognition.

Research on game playing has led to key insights about the nature of intelligence, especially in problem solving.

Problem Solving in Chess

Chess has attracted more attention than other games because it displays these characteristics so clearly:

- It is a complete-information, zero-sum, two-player game with reasonably simple rules.

- It is easy for humans to play chess legally but poorly, but also possible, with experience, to play it very well.

- There are large differences between novice and expert chess players.

As early as 1945, Turing proposed chess as a good domain in which to study computer intelligence.

Chess has also been popular because it is clearly very hard to master, but not as difficult as some games like Go.

Expertise in Chess

Historically, chess masters have often claimed that they play chess well because they are smarter than other people, but:

- de Groot showed that masters and novices carried out the same amount of search and used similar strategies like progressive deepening.

- Chase and Simon showed that master-novice differences in memory for chess board does not apply to random boards (describe experiment).

- Becoming a chess master takes ten years of playing and practice, just as does becoming an expert in other domains like music.

Their conclusion was that masters differ from novices only in knowledge about board patterns and moves appropriate to them.

Drawbacks of Game-Playing Research

However, recent AI research on game playing tells us little about the nature of intelligence because it:

- Ignores what we have learned about game playing in humans.

- Relies on inordinate amounts of search and memory made possible by advances in computing hardware.

- Depends on carefully tuned, domain-specific evaluation functions to guide search that will not work on other games.

Pell (1993) and Epstein (1999) report some of the few efforts that run counter to this trend.

The General Game Playing (games.stanford.edu) competition, now in its Nth year at AAAI, has extended this idea further.

Assignments for Meeting 14
Goal-Driven Problem Solving

Read the articles:

- Langley, P., & Choi, D. (2011). Icarus user’s manual (Technical Report). Institute for the Study of Learning and Expertise, Palo Alto, CA. (Read Section 4) [required]

- Newell, A., & Simon, H. A. (1961). GPS, a program that simulates human thought. In H. Billing (Ed.), Lernede automaten. Munich: Oldenbourg KG. Reprinted in E.A. Feigenbaum & J. Feldman (Eds.), Computers and thought. New York: McGraw-Hill, 1963. [optional]

- Minton, S., & Carbonell, J. G. (1987). Strategies for learning search control rules: An explanation-based approach. Proceedings of the Tenth International Joint Conference on Artificial Intelligence (pp. 228-235). Milan, Italy: Morgan Kaufmann. [optional]

- Examine the fourth exercise (due 11:59 PM on 3/9/2011) and bring questions about it to class.